Remarks

Reconsideration of this Application is respectfully requested.

Upon entry of the foregoing amendment, claims 1 and 49-53 are pending in the application, with claims 1, 49, and 52 being the independent claims. Claim 1 is sought to be amended. No new matter is embraced by this amendment and its entry is respectfully solicited. Based on this amendment and the remarks set forth below, it is respectfully requested that the Examiner reconsider and withdraw all outstanding objections and rejections.

The Applicant would like to thank the Examiner for the interview on January 10, 2002. During that interview the differences between the claimed invention and U.S. Patent No. 5,887,183 to Agarwal *et al.* were discussed. The Examiner acknowledged the differences and indicated that he would look into the case in the next response.

Description of the Invention

The present invention is directed to a method of processing instructions for Single Instruction/Multiple Data (hereinafter, "SIMD") processing. The method includes steps of aligning and ordering vector elements for SIMD processing. To align the vectors for SIMD processing, a first vector is loaded into a first register, and a second vector is loaded into a second register. A first width vector is extracted from the first register and the second register. The resulting extracted vector starts from a first bit of a starting byte in the first width vector and continues through bits of the second width vector. The extracted vector

is then replicated into a third register. To order elements for SIMD processing, a first subset of elements (which may include odd bytes, even bytes, upper bytes, lower bytes or any combination of these) is chosen from the first register and a second subset of elements (which may include odd bytes, even bytes, upper bytes, lower bytes or any combination of these) is chosen from the second register. The chosen elements are replicated into the third register.

An advantage of the present invention is that it allows for faster parallel processing of instructions to be used for SIMD processing. Furthermore, the present invention allows processing, for example, of 64-bit vectors containing different size elements (e.g., a 64-bit vector may contain eight 8-bit elements, four 16-bit elements, two 32-bit elements or one 64-bit element). The vectors are processed according to the element subdivision within each vector register.

Rejections under 35 U.S.C. § 112

In the Office Action dated August 24, 2001, the Examiner has rejected claim 1 under 35 USC § 112, ¶ 2, as being indefinite for failing particularly point out and distinctly claim the subject matter which the Applicants' regard as their invention. Claim 1 is sought to be amended to correct this minor antecedent basis error. In view of this amendment, Applicants respectfully request that this rejection be reconsidered and withdrawn.

Rejections under 35 U.S.C. § 103

In the Office Action dated August 24, 2001, the Examiner has rejected claims 1, 49, and 52 under 35 USC § 103(a) as being unpatentable over U.S. Patent No. 5,887,183 to Agarwal *et al.* in view of U.S. Patent No. 5,922,066 to Cho *et al.*.

With respect to claim 1, the Examiner states that all of the steps of claim 1 except the last the step of replicating of the first width vector into a third register are described by the Agarwal reference and the step of replicating is described by the Cho reference. With respect to claim 49, the Examiner gives a similar rejection under 35 USC § 103(a) and states that even though the Agarwal reference does not specifically describe the writing step of claim 49, the Cho reference provides such step. With respect to claim 52, the Examiner states that the Agarwal reference does not specifically describe the step of selecting the first subset of elements and the step of selecting the second subset of elements, however, the Examiner states that Cho reference provides such steps. The Examiner states that it would have been obvious to one of ordinary skill in the art to combine the two references to produce the claimed subject matter of claim 52.

The Examiner's rejections are respectfully traversed for the reasons set forth below, and the Examiner is respectfully requested to reconsider and withdraw any objections or rejections with respect to claims 1, 49, and 52.

Agarwal describes a method and a system for loading and storing vectors (having a plurality of elements) in a plurality of modes. The vectors are stored in an input storage area, then the elements of the vectors are transferred from the input storage area into a vector

register interface unit. From the vector interface unit, the vector elements are transferred to addressable locations in a preselected pattern in the output storage area. See Agarwal at col. 9, lines 66-67; col. 10, line 1-5; col. 10, lines 46-53; and FIGS. 4A and 4B. The vector elements may be stored in a designated pattern (where such pattern has alternating real and imaginary elements) in the input storage area. Furthermore, the vectors elements may be separated into real and imaginary elements in the output storage area.

Cho describes a system and a method for aligning data for load and/or store instructions. The system of Cho is capable of rotating and shifting data for arithmetic logic instructions. Cho describes an aligner that is adapted to align data as required for load/store instructions and to shift data elements within an operand as required for a shift instruction. Cho uses the same circuit to perform two operations, thus, reducing an overall circuit size. Cho provides for a SIMD processor including an instruction fetch unit, dual instruction decoders, a vector register file, a scalar register file and dual execution units capable of operating in parallel. The instruction decoder of Cho decodes the instructions, which the execution units execute. Such instructions may be to add, subtract, divide or multiply source operands.

The system of Cho is capable of loading/storing data and shifting the data within an operand (which requires a logical or arithmetic operation, left or right shift, rotate or change of position of the elements in the operand). The instruction fetch unit fetches up to two instructions for each of the decoders per each cycle to be processed. The decoders process and pass each instruction onto either the vector or scalar register file. The vector register file contains 64 32-byte vector registers that are organized into two banks of 32 vector registers.

The scalar register file contains 32 32-bit scalar register files, where each register file contains a single 8-bit, 16-bit, or 32-bit value. After processing, the results are forwarded into execution units containing aligners. The execution units perform such functions as load, store and data move operations. The aligner is used in the load and store operations performed by the execution units. For example, for an unaligned load operation, the execution unit requests two data vectors from the memory system, which are two 32-byte data vectors, and a resultant vector is constructed from the two. For an unaligned store, the aligner rotates the data elements in the data storage so that the elements are in their correct positions for storage in two cache lines in the memory system. The aligner has an input select circuit, an element select circuit and a rotation circuit. For data element shifting or rotating, the input select circuit selects one vector operand and another operand, which may be a vector or a scalar, and shifts them into a resultant vector, when the first vector operand is shifted. The shifting and rotating are performed by the aligner include shifting elements by the N byte size elements, where N is an integer representing the number of elements to be shifted or rotated See Cho at col. 6, lines 18-39.

Claim 1

Claim 1 recites the steps of:

> loading a first vector from a memory unit into a first register, wherein the first vector contains a first byte of the aligned vector to be generated;
> loading a second vector from the memory unit into a second register;
> determining a starting byte in the first register wherein the starting byte specifies the first byte of the aligned vector;

extracting the aligned vector from the first register and the second register beginning from the first bit in the starting byte of the first register continuing through bits in the second register; and

replicating the aligned vector into a third register such that the third register contains a plurality of elements aligned for SIMD processing.

As indicated above, the Examiner has acknowledged that Agarwal does not teach the replicating step. In addition, Agarwal does not teach the steps of loading a second vector into a second register, determining a starting byte and extracting an aligned vector.

With respect to the loading step, Agarwal loads only a first vector (i.e., from vector register interface unit 216) into one or two registers (i.e., in vector register array 238). Further, Agarwal does not teach or suggest the claimed step of determining a starting byte in a first register. Instead Agarwal, stores or loads the vectors *consecutively* as they appear either in the input storage area or load area. The vectors may be stored in different patterns, such as reverse order or separated into real and imaginary components. However, Agarwal does not determine which element is the starting byte of the first vector register.

Agarwal also does not teach or suggest the claimed step of extracting the aligned vector from first and second registers beginning with the starting byte. Agarwal merely stores or loads vector register elements to and from addressable locations in preselected patterns. For example, in Agarwal, vector elements are transferred from the vector register interface unit 216 to a vector register in vector register array 238. See FIG. 4A. However, vectors are not taken from first and second registers and no alignment is taking place in the step of writing a vector to a vector register in array 238. Thus, in Agarwal, there is no teaching or suggestion of a step of extracting.

Cho does not provide the missing features. For example, Cho does not teach or suggest at least the steps of determining, and extracting. What Cho describes is a method of shifting and rotating of elements within a vector register, where a shifting operation is performed by shifting an entire first vector operand into another vector register and shifting other elements into that vector register. See Cho at col. 6, lines 18-39. This is different from the present invention, as recited in claim 1, where elements in each vector register are selected by extracting a vector from the first vector register and the second vector register.

Therefore, neither Agarwal nor Cho teaches or suggests every element of claim 1. Furthermore, the combination of Agarwal and Cho does not provide any teaching or suggestion that would lead a person of ordinary skill in the art to create the combination of steps recited in claim 1. While Agarwal discloses a system of loading or storing vectors into registers, the vectors are already pre-aligned before they are stored or loaded. The system in Agarwal determines what type of vectors (stride-1, complex, stride-n or stride-(-1)) are to be loaded/stored and loads/stores them accordingly. Cho merely describes a system of data processing that is capable of shifting and rotating data to be used for load/store functions. The combination of Agarwal and Cho would produce a system capable of performing load and store functions of vectors and performing shift operations to align the vectors from load and store operations. However, the combination would not produce the claimed method including the steps of determining a starting byte, extracting an aligned vector and replicating the aligned vector into a third register for further SIMD processing. Accordingly, Applicants submit that the rejection of claim 1 under 35 U.S.C. §103 is

improper. Based on these remarks, it is respectfully requested that the Examiner reconsider and withdraw the rejection of claim 1.

Claim 49

Claim 49 recites the steps of loading a first source vector into a first register, loading a second source vector into a second register, reading a first plurality of elements from the first register and a second plurality of elements from the second register and writing the first and second plurality of elements into a third register. As discussed above with respect to claim 1, neither Agarwal or Cho teaches or suggests these claimed features. Accordingly, claim 49 is patentable over the combination of Agarwal and Cho for the same reasons discussed above with respect to claim 1. Reconsideration and withdrawal of the rejection of claim 49 is thus respectfully requested.

Claim 52

Claim 52 recites the following steps of loading first and second source vectors into first and second registers, respectively, selecting first subset from the first register (where the subset is selected from odd, even, lower and upper elements) and selecting a second subset from the second register (where the subset is selected from: odd, even, lower and upper elements). Agarwal does not teach or suggest selecting the subsets from each respective register. As discussed above with respect to claim 1, Agarwal describes loading and storing elements in vector registers as they originally appear. The elements might be

subdivided into real and imaginary elements or might be stored in reverse order in resulting vectors. However, claim 52 recites steps of selection of elements from first and second registers to be placed into the third register. These steps are neither taught nor suggested by Agarwal.

As discussed above with respect to claim 1, neither Agarwal or Cho teaches or suggests these claimed features. Accordingly, claim 52 is patentable over the combination of Agarwal and Cho for the same reasons discussed above with respect to claim 1. Reconsideration and withdrawal of the rejection of claim 52 is thus respectfully requested.

Other Matters

The Examiner has rejected the claims 1, 49, and 52 under the doctrine of non-statutory double patenting. To accommodate this rejection, Applicant's are submitting herewith a properly executed terminal disclaimer in accordance with 37 CFR § 1.321(c). In view of this terminal disclaimer, reconsideration and withdrawal of this rejection is respectfully requested.

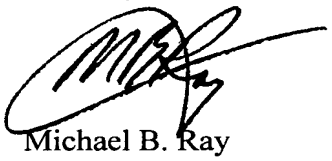
Conclusion

All of the stated grounds of objection and rejection have been properly traversed, accommodated, or rendered moot. Applicant(s) therefore respectfully request(s) that the Examiner reconsider all presently outstanding objections and rejections and that they be withdrawn. Applicant(s) believe that a full and complete reply has been made to the

outstanding Office Action and, as such, the present application is in condition for allowance. If the Examiner believes, for any reason, that personal communication will expedite prosecution of this application, the Examiner is invited to telephone the undersigned at the number provided.

Respectfully submitted,

STERNE, KESSLER, GOLDSTEIN & FOX P.L.L.C.



Michael B. Ray
Attorney for Applicants
Registration No. 33,997

Date: 1/24/01

1100 New York Avenue, N.W.
Suite 600
Washington, D.C. 20005-3934
(202) 371-2600

P108-05.doc

Version with markings to show changes made

1. (once amended) In a computer system including a processor having a plurality of registers, a method for generating an aligned vector of first width from two second width vectors for single instruction multiple data (SIMD) processing, comprising the steps of:

loading a first vector from a memory unit into a first register, wherein the first vector contains a first byte of [an] the aligned vector to be generated;

loading a second vector from the memory unit into a second register;

determining a starting byte in the first register wherein the starting byte specifies the first byte of [an] the aligned vector;

extracting [a first width] the aligned vector from the first register and the second register beginning from the first bit in the [first] starting byte of the first register continuing through [the] bits in the second register; and

replicating the [extracted first width] aligned vector into a third register such that the third register contains a plurality of elements aligned for SIMD processing.